

# Trabajar con ficheros en PHP

<https://www.php.net/manual/es/ref.filesystem.php>

# Trabajar con ficheros en PHP

- Un fichero es una estructura que se guarda en memoria secundaria y que permite guardar información.
- Las operaciones básicas a la hora de trabajar con ficheros son: abrir, leer, escribir y cerrar.

# Trabajar con ficheros en PHP

Algunas funciones con las que se puede obtener información sobre ficheros o directorios son:

- **file\_exists()**: devuelve true si el archivo existe.
- **is\_executable()**: devuelve true si el archivo es ejecutable.
- **is\_readable()**: comprueba si existe un archivo se puede leer.
- **is\_writable()**: comprueba si existe un archivo y se puede escribir en él.
- **is\_file()**: indica si el nombre pasado como parámetro es un archivo.
- **is\_dir()**: indica si el nombre pasado como parámetro es un directorio.
- **filesize()**: devuelve el tamaño de un archivo en bytes.
- **filetype()**: devuelve el tipo de fichero de que se trata: file, dir, block, char, fifo, link o unknown.

# Trabajar con ficheros en PHP

**NOTA:** w y w+ borran el contenido que tuviera el archivo si este existe. Si se quiere crear nuevos archivos y evitar sobrescribir usa a o a+.

## Crear, abrir y cerrar ficheros

- Para crear o abrir un archivo en PHP se usa la función `fopen()`. Si no puede abrir el fichero devuelve `false`.
- A esta función se le pasan dos parámetros, el primero es el nombre del archivo que se quiere abrir (pudiendo indicar la ruta) y el segundo el modo de apertura. Los más habituales son:

r	abre el archivo para lectura, puntero al principio del fichero.	r+	abre el fichero para lectura y escritura, puntero al principio del fichero.
w	abre el fichero en modo sólo escritura, colocando el puntero del archivo al principio. Si el archivo no existe se creará, y si existe su contenido será borrado	w+	Lo mismo que el anterior, pero abre el archivo en modo lectura y escritura.
a	abre el archivo en modo sólo escritura, colocando el puntero del archivo al final del fichero (añade). Si el archivo no existe será creado	a+	Lo mismo que el anterior, pero abre el archivo en modo lectura y escritura

# Trabajar con ficheros en PHP

## Abrir y cerrar ficheros: **fopen()** y **fclose()**

- `fopen()` devuelve un recurso tipo puntero asociado al archivo, un manejador.
- **IMPORTANTE:** *los SO usan diferentes convenciones para los saltos de línea. Por ejemplo, Windows usa `\r\n` como caracteres de final de línea, UNIX usa `\n` y MACintosh usa `\r`. Al abrir esos ficheros con otras aplicaciones aparecerán caracteres extraños si se utilizan los caracteres de fin de línea erróneos al escribir en el fichero. Para evitar incompatibilidades es recomendable usar el **modificador b** como último carácter del parámetro `mode`, que no traducirá la información de los caracteres de fin de línea al abrir el fichero.*
- Una vez hayamos terminado de realizar las operaciones deseadas con el archivo es necesario cerrarlo usando la función **`fclose()`**.

# Trabajar con ficheros en PHP

## Abrir y cerrar ficheros: fopen() y fclose()

```
<?php
/* programa muy sencillo que intenta abrir un fichero creándolo si no
 * existe. Observa que al abrir el fichero usamos el modificador "b" para
 * evitar las incompatibilidades que pueden presentar los diferentes SO
 * con la forma en la que representan el final de línea
 * El archivo se abre en modo w+ por lo que borrará el contenido que tuviera
 */
$archivo = fopen("datos.txt", "w+b");
if( $archivo == false )
    echo "Error al crear el archivo";
else
    echo "El archivo ha sido creado";
fclose($archivo);    // Cerrar el archivo
```

# Trabajar con ficheros en PHP

Si abres el fichero datos.txt (que se encuentra en la misma carpeta del fichero) verás el contenido que se ha escrito en el archivo.

## Escribir texto en ficheros: fwrite(), fputs()

- Para escribir texto en un archivo usaremos las funciones fwrite() o fputs() (esta es un alias de fwrite), las cuales devolverán false en caso de error.
- En caso de éxito devuelve en nº de bytes que se han escrito.

```
// Abrir el archivo, creándolo si no existe:
$archivo = fopen("datos.txt","w+b");

if( $archivo == false ) {
    echo "Error al crear el archivo";
}
else
{
    fwrite($archivo, "Hola ¿Cómo estás?\r\n");
    fwrite($archivo, "Estamos probando\r\n");
    fwrite($archivo, "el uso de archivos ");
    fwrite($archivo, "en PHP ¿Qué te parece?");
    fclose($archivo);
}

// Cerrar el archivo:
echo "Fin del programa";
```

En la primera instrucción de escritura se añade un salto de línea en formato Windows

# Trabajar con ficheros en PHP

## Leer de un fichero: **fread()**

- Devuelve una cadena de texto desde la posición actual del puntero al fichero, con el contenido leído de un archivo.
- En caso de error devuelve false.
- A esta función se le pasan 2 parámetros, el primero la referencia del archivo (que debe estar abierto) del que queremos leer. El segundo parámetro es un valor entero que indica el número de bytes que queremos leer (lee hasta ese número).
- Si necesitamos leer todo el archivo podemos usar la función `filesize()` como segundo parámetro: **`$contents=fread($file,filesize($file));`**.
- Tras usar **`fread()`** el puntero al archivo quedará en la última posición leída.
- *Según la documentación oficial de PHP, en sistemas operativos que diferencian entre archivos de texto y archivos binarios (como Windows) **se recomienda abrir el fichero usando el modificador b.***



# Trabajar con ficheros en PHP

Una vez abierto el fichero:

- Leemos los 13 primeros bytes.
- Leemos los siguientes 14 bytes.
- Movemos el puntero del fichero al comienzo con la función `rewind()`.
- Leemos el contenido completo del fichero

```
// Abrir el archivo en modo de sólo lectura:
$archivo = fopen("datos.txt","rb");
if( $archivo == false ) {
    echo "Error al abrir el archivo";
}
else
{
    /* Leemos un determinado número de caracteres*/
    $cadena1 = fread($archivo, 13);
    $cadena2 = fread($archivo, 14);
    /* Volvemos a situar el puntero al principio del archivo con la función rewind*/
    rewind($archivo);
    /* Leemos hasta el final del archivo*/
    $cadena3 = fread($archivo, filesize("datos.txt"));

    if( ($cadena1 == false) || ($cadena2 == false) || ($cadena3 == false))
        echo "Error al leer el archivo";
    else
    {
        echo "<p>\$cadena1: '".\$cadena1.'"</p>";
        echo "<p>\$cadena2: '".\$cadena2.'"</p>";
        echo "<p>\$cadena3: '".\$cadena3.'"</p>";
    }

    // Cerrar el archivo:
    fclose($archivo);
}
```

# Trabajar con ficheros en PHP

## **Leer de un fichero: `file_get_contents()`**

- Con la función `file_get_contents()` podemos obtener todo el contenido del archivo en una cadena de texto, pudiendo también indicarle la posición inicial a partir de la que deseamos extraer el texto (comenzando desde cero), y el número de caracteres a obtener.
- Devuelve `false` en caso de error.
- Esta función es conveniente usarla si solo queremos poner texto en una cadena, ya que tiene mejor rendimiento que `fread()`.

# Trabajar con ficheros en PHP

```
$cadena = file_get_contents("datos.txt");  
echo "<p>\$cadena después de haber leído todo el fichero es '".\$cadena."'<br/>";  
  
/*  
 * Vamos a leer ahora desde el carácter 3 12 caracteres.  
 * los dos parámetros que tienen valor null, son:  
 *     - una constante que indica el directorio en el que realizar la búsqueda  
 *     - contexto  
 * Ambos parámetros quedan fuera del alcance del curso y por tanto no los usaremos  
 * Devuelve false en caso de error.  
 */  
  
$cadena = file_get_contents("datos.txt", null, null, 3, 12);  
echo "<p>Ahora \$cadena es '".\$cadena."'</p>";
```

# Trabajar con ficheros en PHP

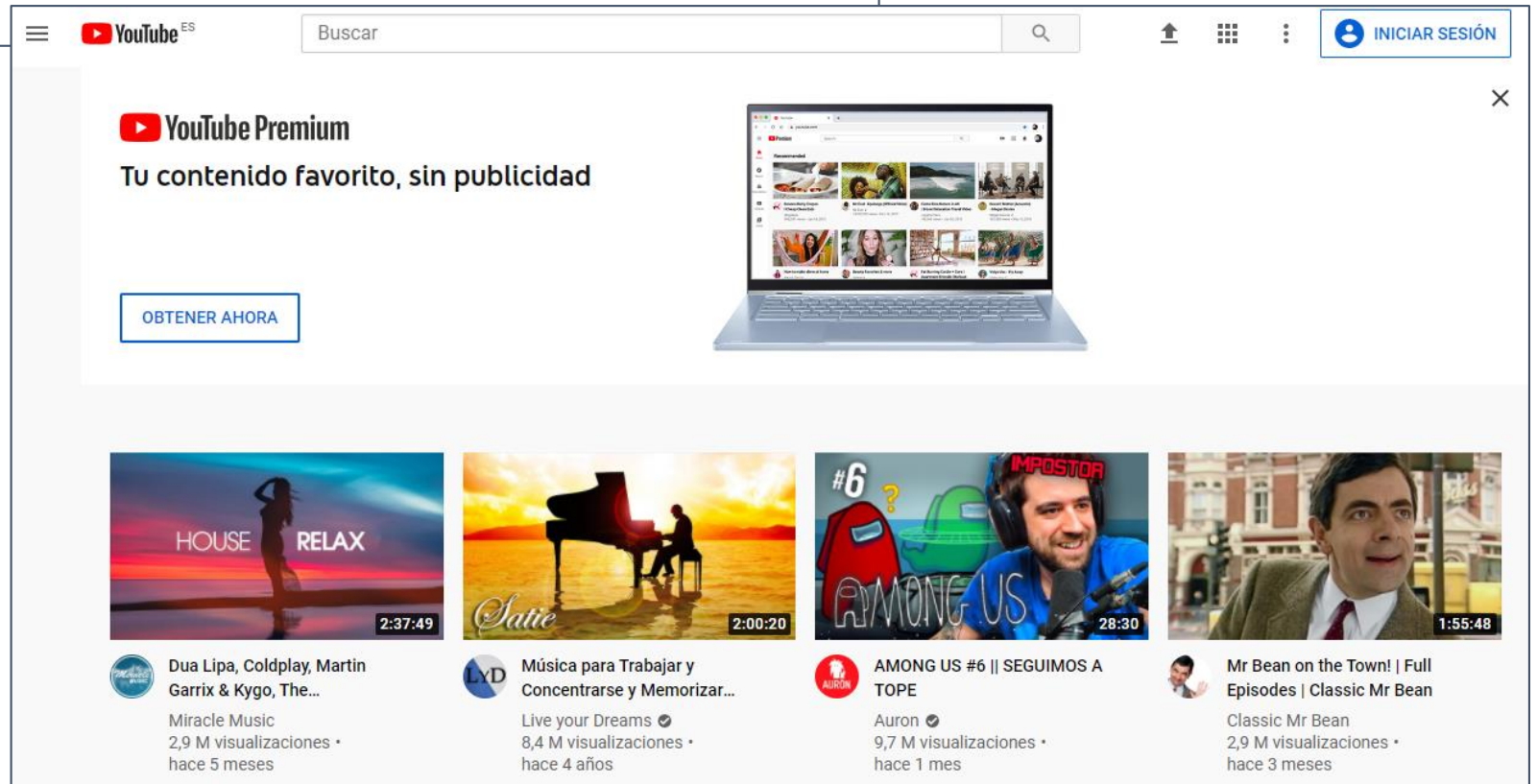
## **Leer de un fichero: `file_get_contents()`**

- Con `file_get_contents()` también podemos obtener el código fuente de una página de un sitio web.
- El navegador "interpreta" lo que lee, que en este caso es código HTML y no muestra el código fuente sino que renderiza el contenido (ten en cuenta que puede haber páginas en las que no se muestren todos los elementos como en la original porque haya información a la que no podamos acceder, bien por rutas relativas, bien por uso de JS, etc.)

# Trabajar con ficheros en PHP

## Leer de un fichero: file\_get\_contents()

```
$pagina_inicio = file_get_contents('https://www.youtube.com/');  
echo $pagina_inicio;
```



# Trabajar con ficheros en PHP

## Leer de un fichero: `file_get_contents()`

- Al tener todo el código en una variable de tipo cadena, podemos usar las funciones de Strings de PHP para hacer, por ejemplo, búsquedas.

```
$pagina = "https://20minutos.es/";
$fileTexto = file_get_contents($pagina);

function buscarH2($texto, $html) {
    $buscaTexto = "<h2";
    echo "La etiqueta h2 aparece: ".substr_count($texto, $buscaTexto).
        " veces en ".$html."<br><br>";
}

buscarH2($fileTexto, $pagina);
```

La etiqueta h2 aparece: 23 veces en https://20minutos.es/

En la variable `$filetexto` tenemos todo el código de la url.

La función `bucarH2` busca en la cadena el número de veces que aparece la etiqueta `"<h2"` (no se incluye el `>` por si hay elementos de estilo, como clases, definidos)

# Trabajar con ficheros en PHP

## Función file()

- Esta función devuelve un array en el que cada elemento corresponde a una línea del fichero.
- En caso de fallo, devuelve FALSE.

```
$aCadena = file("datos.txt");  
print_r($aCadena);
```

```
Array ( [0] => Hola ¿Cómo estás? [1] => Estamos probando [2] => el uso de archivos en PHP ¿Qué te parece? )
```

# Trabajar con ficheros en PHP

## Función fgets()

- Esta función devuelve una cadena con el contenido de una línea del fichero (hasta la marca de salto de línea) o false en caso de error.
- La marca de fin de fichero se detecta con la función feof().

```
$archivo = fopen("datos.txt","rb");

// Recorremos el archivo mostrando el contenido de cada línea
while( feof($archivo) == false )
{
    echo fgets($archivo) . "<br />";
}

fclose($archivo);
```

Hola ¿Cómo estás?  
Estamos probando  
el uso de archivos en PHP ¿Qué te parece?

Observa que usamos el modificador b al abrir el fichero para evitar problemas con los caracteres de fin de línea de los SO.

Leemos del fichero mientras no encontremos la marca de fin de fichero y mostramos cada línea leída.

**RECUERDA:** siempre que usamos la función fopen() hay que usar la función fclose() para cerrar el fichero.



# Trabajar con ficheros en PHP

## **Leer de un fichero: fgetcsv()**

- Para leer ficheros csv (comma-separated values) se utiliza la función `fgetcsv()` indicando el manejador del archivo (una vez abierto), y el carácter separador, que por defecto es el carácter barra invertida.
- Esta función lee una línea entera del fichero y la analiza buscando campos csv.
- Devuelve un array con los campos leídos.
- Una línea en blanco no será tratada como error sino como un único campo null.
- Cuando detecta el fin del fichero devuelve false.

# Trabajar con ficheros en PHP

```
$file = fopen("Datos-alumnos.csv","rb");
if($file){
    echo "Error al abrir el fichero";
}else{
    while(!feof($file)){
        echo "<tr>";
        $linea =fgetcsv($file,1000,';');
        if($linea){
            foreach($linea as $valor)
            {
                echo "<td>$valor</td>";
            }
            echo "<tr>";
        }
    }
    fclose($file);
}
```

Mientras no encontremos la marca de fin de fichero leemos una línea del fichero csv.

Indicamos la longitud de línea (en bytes). Debe ser mayor que la longitud de la línea más larga. Si no se indica el proceso es algo más lento.

El carácter delimitador es el punto y coma.

Mostramos los resultados en una tabla HTML.

# Trabajar con ficheros en PHP

```
MARTA;BLANCO MELENDEZ;634315599;martabl@gmail.com
MARGARITA;MORON SANCHEZ;687177286;margaritamo@gmail.com
MARIA;MORALES RODRIGUEZ;641787645;mariamo@gmail.com
LAURA;JIMENEZ JIMENEZ;640132314;lauraji@gmail.com
SARA MARIA;ARZUMENDI TOCON;649118051;saramariaar@gmail.com
MARIA DE LA SIERRA;ROLDAN GOMEZ;621513631;mariadelasierraro@gmail.com
MIGUEL ANGEL;CARO CRUZ;647143792;miguelangelca@gmail.com
PABLO;FERNANDEZ SANCHEZ;619699166;pablofe@gmail.com
ANA;ESPEJO MORAN;651029704;anaes@gmail.com
```

Fichero csv

Resultado de ejecutar el  
programa en el navegador

MARTA	BLANCO MELENDEZ	634315599 martabl@gmail.com
MARGARITA	MORON SANCHEZ	687177286 margaritamo@gmail.com
MARIA	MORALES RODRIGUEZ	641787645 mariamo@gmail.com
LAURA	JIMENEZ JIMENEZ	640132314 lauraji@gmail.com
SARA MARIA	ARZUMENDI TOCON	649118051 saramariaar@gmail.com
MARIA DE LA SIERRA	ROLDAN GOMEZ	621513631 mariadelasierraro@gmail.com
MIGUEL ANGEL	CARO CRUZ	647143792 miguelangelca@gmail.com
PABLO	FERNANDEZ SANCHEZ	619699166 pablofe@gmail.com
ANA	ESPEJO MORAN	651029704 anaes@gmail.com
JOAQUIN	DEL RIO SANCHEZ-MATAMOROS	654379763 joaquinde@gmail.com
PAULO	CALVO AGUILAR	683888151 pauloca@gmail.com
ELENA	BLAZQUEZ GIL	620469594 elenabl@gmail.com
DIEGO	MORILLA MAZA	631773291 diegomo@gmail.com

Fin